

```

1 import cv2, os, sys
2 import numpy as np
3
4 #*****
5 # 문서 파일 읽어서 opencv 그림객체에 저장하기
6 #*****
7 def read_text_image_file(imageFileName) :
8     DISPLAY_FLAG = 1
9     # 현재 폴더 경로 추출
10    workingDirectory = os.getcwd()
11    argumentFileName = os.path.join(workingDirectory, imageFileName)
12    originalImage = cv2.imread(argumentFileName, cv2.IMREAD_COLOR)
13    imageHeight, imageWidth, imageChannel = originalImage.shape
14    print("Original Image ==> Height:{},Width:{},Channel:{}".format(imageHeight, imageWidth, imageChannel))
15
16    # 입력 문서가 클 경우, 크기 축소
17    if imageWidth > STANDARD_IMAGE_WIDTH :
18        resizeRatio = STANDARD_IMAGE_WIDTH / imageWidth
19        originalImage = cv2.resize(originalImage,(STANDARD_IMAGE_WIDTH, int(imageHeight*resizeRatio)), W
20            interpolation=cv2.INTER_CUBIC)
21
22    # Gray 이미지로 변환
23    grayImageOriginal = cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY)
24    # 글자와 배경색 전환
25    grayImage = 255 - grayImageOriginal
26
27    imageHeight, imageWidth, imageChannel = originalImage.shape
28    print("Resized Image ==> Height:{},Width:{},Channel:{}".format(imageHeight, imageWidth, imageChannel))
29    show_image(originalImage,"ORIGINAL",[0,0])
30    show_image(grayImageOriginal,"GrayOriginal",[200,0],DISPLAY_FLAG)
31    show_image(grayImage,"GRAY",[400,0],DISPLAY_FLAG)
32
33
34    return grayImage
35
36 #*****
37 # OpenCV함수를 이용한 이미지 출력
38 #*****
39 def show_image(imageData,title,position,show=1) :
40     if show != 1:
41         return
42     cv2.namedWindow(title,cv2.WINDOW_NORMAL)
43     cv2.moveWindow(title,position[0],position[1])
44     cv2.imshow(title, imageData)
45
46 #*****
47 # 모폴로지 연산: gray 이미지로부터 이진 이미지로 변환 , 수평 Smearing, 윤곽추출하여 리턴
48 #*****
49 def do_morphology_and_find_contours(imageData) :
50     DISPLAY_FLAG = 1
51     SHOW_MORPHOLOGY_FLAG = 1
52
53     kernelElement = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
54
55     # Dilation 여러번 수행
56     if SHOW_MORPHOLOGY_FLAG:
57         dilatedImage = imageData.copy()
58         for n in range(5):
59             dilatedImage = cv2.dilate(dilatedImage,kernelElement)
60             windowTitle = "DILATED_"+str(n+1)+" th"
61             if DISPLAY_FLAG :
62                 show_image(dilatedImage>windowTitle,[100*n,200])
63
64     # Morphology 간단한 방법
65     # 1. Opening : Erosion 이후 Dilation
66     openImage = cv2.morphologyEx(imageData, cv2.MORPH_OPEN, kernelElement, 5)
67     show_image(openImage,"OPENING 5 Times",[0,400],DISPLAY_FLAG)
68     # 2. Closing : Dilation 이후 Erosion
69     closedImage = cv2.morphologyEx(imageData, cv2.MORPH_CLOSE, kernelElement, 5)
70     show_image(closedImage,"CLOSING 5 Times",[100,400],DISPLAY_FLAG)
71
72     # 윤곽추출 : Dilation결과 - Erosion 결과
73     gradientImage = cv2.morphologyEx(imageData, cv2.MORPH_GRADIENT, kernelElement)
74     show_image(gradientImage,"GRADIENT",[200,400],DISPLAY_FLAG)
75
76     # 이진화
77     _, binaryImage = cv2.threshold(gradientImage, 0.0, 250.0, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
78     show_image(binaryImage,"BINARY",[300,400],DISPLAY_FLAG)
79
80     # 수평방향 smearing 하기

```

```

81 kernelElement = cv2.getStructuringElement(cv2.MORPH_RECT, (15,6)) # 가로방향 커널 생성
82 connectedImage = cv2.morphologyEx(binaryImage, cv2.MORPH_CLOSE, kernelElement) # CLOSE 모폴로지 연산 실행
83 show_image(connectedImage, "H-SMEARED", [400,400])
84
85 # 가로방향으로 smearing된 edge영상에서 연결된 테두리(contour) 다시 찾기
86 # using RETR_EXTERNAL instead of RETR_CCOMP
87 contours, hierarchy = cv2.findContours(connectedImage, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
88
89 return contours
90
91 #####
92 # 텍스트 라인 찾기
93 #####
94 def find_text_lines(grayImage, contours) :
95
96     #blankImage = np.zeros((height,width), dtype=np.uint8)
97     # Color 이미지로 변환
98     colorImage = cv2.cvtColor(grayImage, cv2.COLOR_GRAY2BGR)
99
100     for idx in range(len(contours)):
101         x, y, w, h = cv2.boundingRect(contours[idx])
102         x2 = x+w; y2=y+h
103         #center_y = int(y+h/2+0.5)
104         #center_x = int(x+w/2+0.5)
105         cv2.rectangle(colorImage, (x, y), (x+w, y+h), (0,255,255),2) # yellow
106
107         #-----
108         # 글자 영역 MBR 인지 확인
109         #-----
110         if idx == 0:
111             prev_center_y = prev_center_x = 0
112             prev_x2 = prev_y2 = 0
113
114         area_size = w * h
115         printed_size = cv2.countNonZero(grayImage[y:y+h, x:x+w])
116         r = float (printed_size / area_size)
117         print ('r= {0:.3},{1},{2}'.format(r,w,h))
118         if r >= 0.15 and w >= 2 and h >= 8 and h < 150 :#and w/h > 0.05 and w < 1000 :
119             cv2.rectangle(colorImage, (x, y), (x+w, y+h), (0,255,0), 2) # green
120         else:
121             cv2.drawContours(colorImage, contours, idx, (0, 0, 255), -1) # red
122
123     show_image(colorImage, "TEXT_MBR", [500,500])
124 #####
125 # Main
126 #####
127 # 전역변수 설정
128 STANDARD_IMAGE_WIDTH = 2500
129
130 if __name__ == "__main__":
131     while True:
132         print("To finish the program, strike 'ESC' key on the image window !!!")
133         imageFileName = input("Input Math_Image FileName: ")
134
135         if len(imageFileName) > 1 :
136             grayImage = read_text_image_file(imageFileName)
137             contours = do_morphology_and_find_contours(grayImage)
138
139             find_text_lines(grayImage, contours)
140             '''
141             recog_texts()
142             post_process()
143             print_result()
144             '''
145             hitKey = cv2.waitKey(0)
146             if hitKey == 27 :
147                 break
148         else :
149             print("Bye...")
150             break
151     cv2.destroyAllWindows()

```